IBM Security

Outdated encryption standards pose a serious risk of data breach

A security update on Internet encryption

IBM X-Force® Research
Managed Security Services Report

IBM

Click here to start ▶

Contents

Executive overview

Many businesses today depend upon encryption standards that date back to the late 1990s. These standards are now considered insufficient for protecting sensitive, confidential or private data according to the National Institute of Standards and Technology[1]. Over the last many months, we have seen security researchers publish information disclosing various weaknesses in these standards. Highly publicized vulnerabilities such as BEAST, POODLE, POODLE V2, Log Jam, Bar Mitzvah, STORM and FREAK reduce the pool of encryption standards known to be still viable. We wouldn't be surprised if similar publicized vulnerabilities continue until the shortcomings of every aspect of these old encryption standards are fully exposed and they are retired.

About this report

This IBM® X-Force® Research report was created by the IBM Managed Security Services Threat Research group, a team of experienced and skilled security analysts working diligently to keep IBM clients informed and prepared for the latest cybersecurity threats. This research team analyzes security data from many internal and external sources, including event data, activity and trends sourced from thousands of endpoints managed and monitored by IBM.

Contents

As an industry (consumers, IT organizations and vendors of technologies), we have the means and the responsibility to get ahead of these threats. Every IT organization and every vendor should have an aggressive strategy and an active project in place to move away from the old encryption standards and to adopt the newer standards that have been published and available since 2008. At some time in the future, we expect that these standards too will become unsafe, which could expose IT organizations to a scramble that will be far worse than the reaction to the POODLE vulnerability. That's why IT organizations should establish an ongoing process for staying current with encryption standards as new ones become available.

The purpose of this paper, which focuses on encryption standards related to data in transit, is twofold. First, it provides high-level information needed to help make sense of the issues created by the use of old encryption standards. Second, it provides recommendations for building a strategy for adopting stronger encryption standards, which will help reduce a company's security risk and the effort spent patching its systems.

Every IT organization and vendor should have a plan in place to move away from old encryption standards.

Contents

Key technology aspects of securing data in transit

Protocol encryption standards for the Internet. The Internet Engineering Task Force (IETF) defines the protocol standards for securing data transmitted across the Internet. This standard, called Transport Layer Security (TLS) Protocol, has multiple versions. We will address which versions of TLS are nearing their end of life and which version is considered safe (for now).

Algorithm encryption standards for the protection of data. The IETF specifies algorithms that are used for encrypting data, whether those data are being transmitted (in transit) or stored (at rest). The TLS standard includes a number of these algorithms. Key elements to understand are:

- **Cipher suites.** These are sets of cryptographic algorithms, the combination of which control how the data in transit is secured. We will be discussing which cipher suites should be avoided and which should be preferred.
- **Digital certificates.** A secured website uses a digital certificate that identifies the site. The more reputable websites use a digital certificate that has been signed by a trusted authority that validates the site is what it claims to be. This trusted authority can be one of the few dozen recognized by the browser creator, or a company may choose to create its own certificate authority and place its own trusted root certificate inside the browsers' certificate stores. This is typically done to identify and secure intranet sites. The key security elements used in the certificate are the encryption algorithm, the key size being used to encrypt the data, and the signing algorithm. We will discuss key sizes and which algorithms should be avoided and which should be preferred.

Standards for transport protocols and data encryption both help protect data in transit across the Internet.

Contents

A brief history of protocol encryption standards for the Internet

The biggest challenge related to protecting data transmitted across the Internet is that the industry has relied heavily on outdated protocol encryption standards that date back to the 1990s. The SSL (Secure Socket Layer) protocol standard was designed to support the original web servers on the nascent Internet. The SSL protocol standard had three versions: SSLv1, SSLv2 and SSLv3. The last version of the SSL protocol, SSLv3, was published in 1996. All variations of SSL are now considered unsafe, and the POODLE vulnerability was the final nail in the SSL protocol's coffin.[2]

The SSLv3 standard was replaced with a new standard called TLS (Transport Layer Security). TLS 1.0 was published in 1999. Even TLS 1.0 is showing its age. Unfortunately, over the years, the industry as a whole has been heavily dependent on SSLv3 and TLS 1.0.

TLS 1.1 was published in 2006, and in 2008, a significant update to the TLS standard was published called TLS 1.2. However, adoption of TLS 1.2 by both server and client technologies has been slow, at least until recently. The primary challenge for adoption of TLS 1.2 has been around interoperability; both sides of the connection need to support TLS 1.2.

Over the last few years, the industry has progressed well in moving to TLS 1.2. Most server-side vendors have been adding support for TLS 1.2, giving IT organizations and consumers the means to use the stronger set of encryption standards covered under the umbrella of TLS 1.2. For consumers, we have seen major browser vendors include TLS 1.2 in their latest versions. For consumers to take advantage of these stronger encryption standards, they now need to upgrade to current browser versions. For more information on TLS 1.2 adoption for browsers, see *Template: TLS/SSL support history of web browsers.*[3]

Older (unsupported) versions of operating systems, middleware or hardware may not support TLS 1.2. Generally, these OS versions are no longer being supported by their creators and should be viewed as end-of-life products. Continuing to use these older OS versions, which only include old encryption standards, could expose corporate and customer data to breaches.

Contents

Why old encryption protocols and standards have become unsafe

As mentioned, many businesses are relying on encryption standards that date back to the late 90s. Since then, computer processors and networks have become much faster and are able to run multiple tasks in parallel. Older algorithms based their security (in part) on the principle that the amount of processing power required to use a brute force attack (trying every key combination to decrypt data) against the algorithm would be beyond the capabilities of the attackers. This design assumption is no longer realistic. In addition, security researchers have discovered inherit design flaws as well as finding implementation flaws in the encryption protocols and algorithms themselves.

TLS protocol versions that are viable and versions considered unsafe

Secure Socket Layer (SSL)

The POODLE vulnerability effectively demonstrated that SSLv3 and earlier versions of the SSL protocol are no longer safe to use.[4] Based on our experience, many IT organizations are rushing to ensure SSLv3 (and earlier) is disabled for both internal and external network connections.

Transport Layer Security (TLS) 1.0

While we have not seen full agreement across the industry for TLS 1.0 to be added to the unsafe list, there is a lot of debate on its viability and soundness as outlined by the National Institute of Standards and Technology[5] and the SANS Institute[6]. We have seen a number of security vulnerabilities that have resulted in patching aspects of TLS 1.0 or have caused retirement of some of the optional encryption algorithms leveraged in TLS 1.0. Security researchers continue to focus their research on TLS 1.0, publishing information every few months on further problems found with this standard. These researchers have further reduced the pool of safe encryption algorithms in TLS 1.0, and it's only a matter of time before the industry views TLS 1.0 as no longer safe to use. Many security experts today would argue we have already reached the breaking point for TLS 1.0.

A number of IT organizations are additionally concerned about TLS 1.0 because some of the recent vulnerabilities require client-side security fixes. Those IT organizations are uncomfortable relying on their users and customers to patch their systems, especially when the IT organizations are ultimately responsible for the integrity and confidentiality of any data being transmitted over the Internet.

Contents

Furthermore, portions of the industry are publishing recommendations to stop using TLS 1.0. For example the Payment Card Industry (PCI) standard has declared TLS 1.0 unsafe for payment transactions and set an initial date of June 2016 to upgrade all software and hardware.[7] For more information, see the PCI Security Standards Council website. Additionally, the National Institute of Standards and Technology, in its NIST Special Publication 800-52 Revision 1,[8] is recommending migration to TLS 1.2.

Should TLS 1.0 be found to have a catastrophic weakness, we could see the industry get into a scramble that could be far worse than the one to disable SSLv3 following the POODLE vulnerability—especially given the wide use of TLS 1.0 compared to SSLv3.

Transport Layer Security (TLS) 1.1

While TLS 1.1 is an improvement over TLS 1.0 (for example, its resistance to BEAST), this standard has some additional inherent weaknesses. TLS 1.1 RFC acknowledges attacks on CBC mode that rely on the time to compute the message authentication code (MAC).[9] We may not be surprised if further weaknesses would be found, which, too, could send the industry into another struggle to disable yet another older protocol before attackers take advantage.

Transport Layer Security (TLS) 1.2 and the upcoming 1.3

TLS 1.2 is currently considered the preferred version, as it is the most viable TLS version available today according to the National Institute of Standards and Technology.[10] The standards community is in the process of developing the next version, called TLS 1.3, which is currently in draft status.[11] At the time of writing this paper, it is not known when this new standard will be available. Based on the adoption time frame seen for TLS 1.2, we expect that when this new standard is available, broad adoption will take years.

Contents

Cipher suites that are not considered safe

The discovery of the FREAK, Bar Mitzvah and LogJam vulnerabilities has further reduced the pool of algorithm encryption standards for TLS. Many vulnerable algorithms are leveraged in TLS 1.0, underscoring the unsuitability of the protocol. Given the frequency and the number of security vulnerabilities that have emerged against SSL and TLS since 2014, we expect this to continue as computational power increases, facilitating brute force decryption. It is likely that further cryptographic weaknesses in algorithms will be discovered.

Some of the cipher suites that are recognized in TLS 1.0 are no longer safe and no longer recommended. For example, the FREAK[12] vulnerability was uncovered in RSA_EXPORT ciphers. Similarly, the Bar Mitzvah[13] vulnerability showed that the RC4 cipher is no longer safe to use.

Other cipher suites have had standardized patching. For example, the Log Jam[14] and BEAST[15] vulnerability disclosures required security patches be delivered for Diffie-Hellman key exchanges and CBC ciphers as part of TLS 1.0. If you are using these cipher suites, be sure to keep your patching up to date.

Security researchers have already demonstrated that using SHA1 (Secure Hash Algorithm 1) as a signing algorithm for certificates is unsafe according to security researchers Marc Stevens, Pierre Karpman, and Thomas Peyrin.[16] Some can argue that it is only a matter of time before additional security research will render SHA1-based ciphers obsolete. Ciphers using AES (Advanced Encryption Standard) or SHA2 or higher digital signature algorithms are still considered viable.

Digital certificates that are not considered safe

As outlined by National Institute of Standards publication 800-57,[17] digital certificates that are generated using an RSA or DSA key size less than 2048 bits, using an EC key size less than 384 bits, or using an MD5 or SHA1 signing algorithm are all considered unsafe. Each of the major browser vendors plans on enhancing their browsers to block any website using these weak standards. What's more, the identity services company GlobalSign is phasing out SHA1 hashes in favor of the newer SHA256.[18]

Contents

Recommendations for data in transit encryption

The recommendations that follow are general recommendations only. Every environment is different and each reader should assess these recommendations against their specific environment. These recommendations are intended to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures.

Use TLS 1.2. Technology vendors and IT organizations should have in place today a strategy and active projects to adopt the TLS 1.2 standard for both internal and external network connections. Fifty-five percent of all data breaches occur inside a company's network, according to the *IBM 2015 Cyber Security Index* report.[19] As a means of transition, an IT organization can consider setting up a tolerant configuration that supports TLS 1.0, 1.1 and 1.2 that allows for the strongest protocol encryption version to be used per the TLS specifications. The TLS standard contains the concept that the highest protocol encryption version will be used that both the client and server support. Once all network connections are able to support TLS 1.2, then the acceptance of TLS 1.0 and 1.1 protocols should be disabled.

Require strong cipher suites. We recommend using cipher suites that support a minimum of AES 128 with SHA2 or higher. If you're using Diffie-Hellman (DH) ciphers, ensure you have applied patches or fixes for CVE-2015-2808 and that you're using a key size of at least 2048 bits. AES-GCM ciphers should be used instead of CBC (Cyber Block Chaining)-based ciphers.

The TLS standard supports a "tolerant configuration," meaning the strongest encryption that is supported by both sender and receiver will be used to transport data.

Contents

Use minimum RSA or DSA 2048 key size and SHA2 signing algorithm and encryption ciphers for digital certificates. IBM recommends using a minimum of 2048-bit keys and using SHA2 or higher and encryption ciphers of 384 bits or higher for digital certificates.

Keep current on software and hardware. New fixes are released regularly to address vulnerabilities and weaknesses in encryption algorithms. It's critical to stay current with security fixes, applying them as soon as possible.

In situations in which TLS 1.0 and 1.1 or weaker ciphers must be used, we recommend the following:

- Stay current with all supported vendor releases, patches and software maintenance levels for both software and hardware.
- Avoid old hardware or software that is only available on an extended support contract, as new encryption standards may not be included or back-ported to the old hardware or software.
- Do not use any encryption ciphers based on RC4.
- Do not use any encryption ciphers based on RSA_EXPORT.
- Do not use any DES (Data Encryption Standard)-based ciphers; instead, use 3DES or AES (Advanced Encryption Standard) ciphers.

Recommendation for data at rest

DES encryption or SHA1 hashing algorithms are also insufficiently secure for protecting data at rest (that which exists in online or offline storage). IBM's recommendation is to move toward stronger algorithms such as AES 128 or higher, and using SHA2 or higher for hashes.

Summary

This paper is intended to serve as a wakeup call to all IT organizations. Security researchers continue to focus on finding flaws in old encryption standards for data in transit, in particular TLS 1.0 and earlier standards. In order to help minimize the business risk of data breaches and to hopefully avoid the next POODLE scramble, it is critical that every IT organization has in place a strategy and an active project to adopt newer encryption standards, TLS 1.2 and SHA2 in particular.

Contents

Protect your enterprise while reducing cost and complexity

From infrastructure, data and application protection to cloud and managed security services, IBM Security Services has the expertise to help safeguard your company's critical assets. We protect some of the most sophisticated networks in the world and employ some of the best minds in the business.

IBM offers services to help you optimize your security program, stop advanced threats, protect data and safeguard cloud and mobile. Our Data Loss and Encryption Services utilize industry-leading encryption and software to help you protect data that is accessed, stored and transmitted on your endpoint devices. The Critical Data Protection Program helps you identify, define and protect the data that is most important to your organization. Managed Data Protection Services for Guardium® provide the skilled resources you need to help you manage your security solution while helping you reduce costs and improve your overall security posture.

About IBM Security

IBM Security offers one of the most advanced and integrated portfolios of enterprise security products and services. The portfolio, supported by world-renowned IBM X-Force research and development, provides security intelligence to help organizations holistically protect their people, infrastructures, data and applications, offering solutions for identity and access management, database security, application development, risk management, endpoint management, network security and more. IBM operates one of the world's broadest security research, development and delivery organizations, monitors billions of security events per day in more than 130 countries, and holds more than 3,000 security patents.

Contents

About the author

Bill O'Donnell is Senior Technical Staff Member, IBM Cloud Middleware, Chief Security Architect, and Chief Security Compliance Officer working in IBM's software product development. Bill is responsible for the security architecture in IBM middleware on-premises and cloud offerings. He is also responsible for a number of security initiatives across the IBM organization, Bill has over 25 years of experience in large-scale mainframe and distributed systems with a unique security focus on software architecture and infrastructure architecture. Bill specializes in end-to-end infrastructure and application security. He has published a number of IBM Redbooks® and papers, and is the author of the book *Secrets of SOA*.

Contributors

Michael Gray – Lead Developer of IBM Global Security Kit (GSkit)

Audrey Timkovich – Lead Developer of IBM JSSE (Java Secure Socket Extension)

Martin Lansche – World Wide Security Lead for IBM Software Services for WebSphere®

Brad Harris – Advanced Security Researcher from IBM X-Force Research

Robert Freeman – Senior Manager of IBM X-Force Research

Ivan Milman – Senior Technical Staff Member, Security and Governance Architect

Chris Poulin – Research Strategist, IBM X-Force

Warren Grunbok – Systems Group Security Architect

Ron Craig – Program Manager, IBM Secure Engineering

Lisa Bradley – Program Manager IBM Product Security Incident Response Team (PSIRT)

Peter Allor – Senior Cyber Security Strategist

Contents

For more information

To learn more about the IBM Security portfolio, please contact your IBM representative or IBM Business Partner, or visit:
ibm.com/security

For more information on security services, visit:
ibm.com/security/services

Follow @IBMSecurity on Twitter or visit the IBM Security Intelligence blog

[1] *http://csrc.nist.gov/publications/nistbul/itlbul2014_04.pdf*

[2] *https://exchange.xforce.ibmcloud.com/vulnerabilities/97013*

[3] *https://en.wikipedia.org/wiki/Template:TLS/SSL_support_history_of_web_browsers*

[4] *https://exchange.xforce.ibmcloud.com/vulnerabilities/97013*

[5] *http://csrc.nist.gov/publications/nistbul/itlbul2014_04.pdf*

[6] *https://www.sans.org/reading-room/whitepapers/analyst/critical-security-controls-guidelines-ssl-tls-management-35995*

[7] *https://www.pcisecuritystandards.org/documents/Migrating_from_SSL_Early_TLS_Information%20Supplement_v1.pdf*

[8] *http://www.nist.gov/itl/csd/tls-043014.cfm*

[9] *https://www.ietf.org/rfc/rfc4346.txt*

[10] *http://csrc.nist.gov/publications/nistbul/itlbul2014_04.pdf*

[11] *https://tools.ietf.org/html/draft-ietf-tls-rfc5246-bis-00*

[12] *https://exchange.xforce.ibmcloud.com/vulnerabilities/99707*

[13] *https://exchange.xforce.ibmcloud.com/vulnerabilities/101851*

[14] *https://exchange.xforce.ibmcloud.com/vulnerabilities/103294*

[15] *http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3389*

[16] *https://marc-stevens.nl/research/papers/KPS_freestart80.pdf*

[17] *http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57Pt3r1.pdf*

[18] *https://support.globalsign.com/customer/portal/articles/1447169-sha-256-rollout*

[19] *http://www-03.ibm.com/security/data-breach/2015-cyber-security-index.html*

Contents

IBM Security

IBM Corporation
IBM Security
Route 100
Somers, NY 10589

Produced in the United States of America
March 2016

IBM

SEL03088-USEN-00